Embedding labeled graphs into occurence matrix

Nicolas Sidère, Pierre Héroux, Jean-Yves Ramel

Embedding labeled graphs into occurence matrix

Nicolas Sidère [1,2] and Pierre Héroux[2] and Jean-Yves Ramel [1]

[1] *Université François Rabelais Tours, Laboratoire Informatique de Tours, 64 Avenue Jean Portalis, 37200 Tours, FRANCE*
E-mail : {nicolas.sidere, ramel}@univ-tours.fr
[2] *Université de Rouen,LITIS EA 4108, BP 12, 76801 Saint-Etienne du Rouvray, FRANCE*
E-mail : pierre.heroux@univ-rouen.fr

Abstract

This paper presents an approach to the classification of structured data with graphs. We suggest to use a graph signature in order to solve the problem of complexity in measuring the distance between graphs. We choose a numerical matrix to embed both topological and labeling information. The use of this matrix allows us to use some classical tools of classification, computation of distances or feature selection. After a description of the matrix and the method to extract it, we compare the results achieved on public graph databases for the classification of symbols and letters using this graph signature.

Keywords : Graph classification, Graph embedding, Feature selection

1 Introduction

Since the start of works on pattern recognition, in the late seventies, the graph theory has known an impressive interest in classification or retrieval. The powerful expressivity associated with a certain smartness have claimed for the development of many tools in recognition or classification of structured data ([1]). The graphs are widely used in real-life applications to represent all sorts of structures (i.e. molecules, technical drawings, web pages, flow charts). In most of these applications, the aim is to recognize an object discribed by a graph.

This pattern recognition scheme is often based-on the computation of a distance between two graphs. The lower the distance is, the more the graphs can be considered as similar. The graph classification is then performed through the nearest neighbour techniques. Several kinds of distance may be used, but the edit distance defined as the minimal cost needed to transform a graph into an other one is a reference. Other distances may be computed through the search of the maximal common subgraph. The computation of these algorithms is NP-Complex [2]. Several works are presented in [1]. Therefore, many algorithms have been proposed aiming at reducing the computation needs [3] or [1].

Among them, an interesting approach is to embed a part of the information carried by the graph into a vectorial feature space, for example a numerical matrix. The study of the differences (or similarities) between two graphs is reduced to a more simple computation between two vectors in an euclidean space. Then, most of the computation time consists in extracting this vectorial description, but this can be performed once for all as a preprocessing of the database. More, this numerical matrix allows the use of tools like feature selection. This selection reduce the size of the matrix by deleting some disturbing elements, so the representativity is more accurate and the extraction quicker.

In this article, the graph embedding approach is presented in the next section. Then, the section 3 deals with some results obtained on public database of graphs issued from letter and symbol graphics with the study of the influence of a feature selection . Finally, section 4 concludes the paper and proposes some extensions.

2 Graph representation using occurence matrix

The basement of the matricial representation of a graph is a lexicon of graphs called *patterns* in the paper. The relevance of the embedding is determined by this lexicon. More, as we want the representation scheme to be

as generic as possible, it is preferred to use a lexicon independent from the database content. However, this lexicon must be comprehensive enough to ensure that it allows to discriminate a graph from another.

We have therefore decided to take as a baseline the non-isomorphic graphs network presented in [6]. The network presents all graphs composed of n edges up to N (N is the maximum number of edges). This network is built iteratively from a graph pattern made up of a single vertex. At each iteration, it is possible to build a pattern of rank n by adding an edge to a pattern of rank $n-1$ with the ability to add a vertex if needed. All solutions are considered. This results in a network complete. A pattern with rank n built from a pattern with rank $n-1$ is called successor. Conversely, the pattern of $n-1$ is called predecessor. A pattern of this network may have several successors. Similarly, several patterns with rank $n-1$ can rise to a single successor. Ways of construction of this non-isomorphic graph network can be stored to build all predecessors and successors of a graph.

The figure 1 shows an example of the construction of such a network till the fourth rank giving a lexicon of 11 patterns. The dotted arrows indicate the path of construction of the network, the arrows are directed from the predecessors towards the successors.

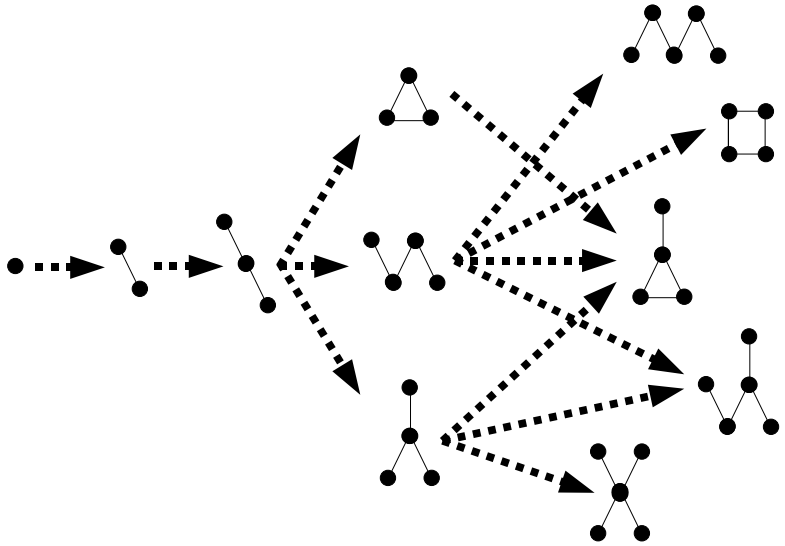

Figure 1: The non-isomorphic graph network

The extraction is the most complex phase of the method. The indexation of the database is done during a preprocessing period. This complexity is directly dependent of the number of patterns. However, the more the size of the lexicon increases, the bigger the patterns it integrates are. The vectorial representation then integrates more information on topology. As the number of patterns increases exponentially with the rank, it is necessary to find a trade-off between expressiveness and complexity.

The matricial representation is built by counting the occurences of each pattern from the lexicon. Then, a numerical signature is obtained. This first row of the embedding represents the topology information of the graph. At this point, the vectorial representation embeds only some topological information and needs to be enriched with encapsulating the labels. We decided to work on multi-labeled graphs, i.e. graphs with labels on vertices and nodes. Each of these labels can be composed with several attributes. The inclusion of these labels is done in four steps :

1. The first step is to list all the labels which occurs at least once in the database, for the vertices and for the nodes. At the end of this step, there are as many lists as the number of types of labels.

2. The second step is to discretize numerical labels. As the vector is based on the frequency of appearance of patterns, only nominal labels are considered. The discretization is done the simplest way by splitting the numerical interval in n classes. Then, the new labels are affected to the graph.

Pattern	.	—	— —	⟍	— — —	⋀
Freq.	4	4	4	2	1	1
A1, corner	2	4	7	4	2	2
A2, corner	0	0	0	0	0	0
A3, corner	0	0	0	0	0	0
A1, endpoint	0	0	0	0	0	0
A2, endpoint	1	2	3	2	1	1
A3, endpoint	1	1	2	2	0	1
X1, Y1	0	3	6	5	2	2
X1, Y2	0	0	0	0	0	0
X2, Y1	0	1	2	1	1	1
X2, Y2	0	0	0	0	0	0

Table 1: Matricial representation of the graph 2

3. The third step is the computation of all possible combinations of labels for a vertex if it is caracterized by several attributes. The same is done for edges.

4. The fourth step is to affect to each topological pattern a vector of possible vertices and edges.

The lexicon now includes labels and can be represented by a matrix. Each column corresponds to a pattern from the lexicon and is then relative to the topological information. Each line is relative to a label combination.

The construction of the vectorial representation can now be performed. The construction of the representation consists on filling all the cells of the table generated with the frequency of each patterns and each labels for this pattern.

The table 1 presents an example of the matricial representation for the graph represented on Fig. 2. We consider that each vertex is labeled with two attributes A and $Type$ such as $A = \{A1, A2, A3\}$ and $Type = \{Endpoint, Corner\}$. Edges also have two attributes X and Y with $X = \{X1, X2\}$ and $Y = \{Y1, Y2\}$.

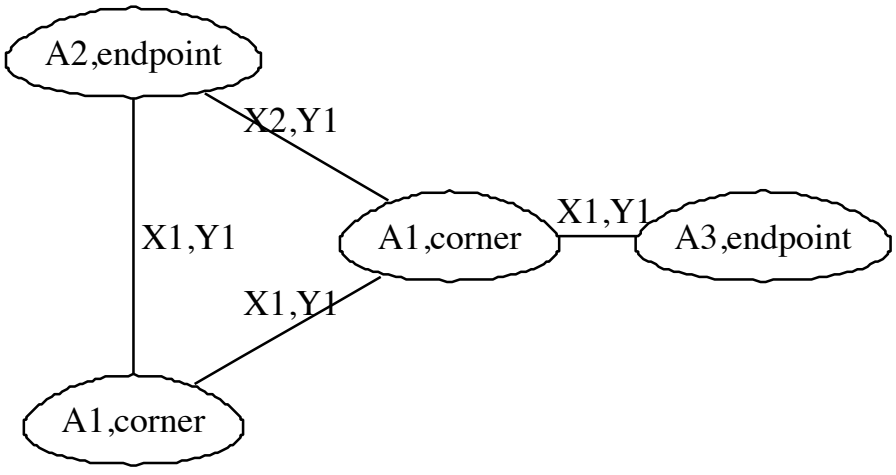

Figure 2: A simple labeled graph

3 Experiments

In this section, the first experiments we conducted for graph classification using the proposed vectorial description are presented. Theses works were leaded on three public bases available at this url : `http://www.iam.unibe.ch/fki/databases/iam-graph-database`. The complete database descriptions are given in [9]. The author of [9] decided to divide the dataset into three disjoint subsets i.e. training, validating and testing. They published result obtained from a k-nearest neighbor classifier (k-NN) is used with graph edit distance. For each dataset, the value of the hyperparameter k is optimized thanks to the validation subset. In [15], the results presented highlights that, depending on the data sets, the classification rate reached with the matricial representation were either equivalent or near with the complexity of computation reduce to a linear time.

As the representation is a numerical matrix, feature selection will be applied to the data. In fact, many patterns or combination of labels does not occur in any graphs. Considering that the lexicon was built totally independantly from the data, it seems rational. At the other hand, some occurence features of the matrix can be non-representative of the class because of its unstability between all graphs from the same class. The feature selection will lighten the matrix in order to decrease the extraction time and even increase the recognition rate by disabling some unused or disturbing feature of the matrix. The chosen feature selection is done in 2 steps. The first one is to evaluate the worth of a subset by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. More details are in [14]. The second step is to search the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point). All the parameters are set with the validation subset of the database and the results given are computed on the test subset.

The next subsections will present the two databases as they are introduced in [9]. Then some classification rates conducted without and with a feature selection are given and compared to those obtained with a graph edit distance, which can be considered as the reference.

3.1 GREC Database

The GREC data set consists of graphs representing symbols from architectural and electronic drawings. The images occur at five different distortion levels. In Fig. 3 for each distortion level one example of a drawing is given. Depending on the distortion level, either erosion, dilation, or other morphological operations are applied. The result is thinned to obtain lines of one pixel width. Finally, graphs are extracted from the resulting denoised images by tracing the linesfrom end to end and detecting intersections as well as corners. Ending points, corners, intersections and circles are represented by nodes and labeled with a two-dimensional attribute giving their position. The nodes are connected by undirected edges which are labeled as line or arc. An additional attribute specifies the angle with respect to the horizontal direction or the diameter in case of arcs. From the original GREC database [10], 22 classes are considered. For an adequately sized set, all graphs are distorted nine times to obtain a data set containing 1,100 graphs uniformely distributed over the 22 classes. The resulting set is split into a traininig and a validation set of size 286 each, and a test set of size 528. The classification rate achieved by the author of [9] on this data set is 95.5%.

Graph edit distance	294 patterns	42 patterns
95.50 %	95.83 %	97.53 %

Table 2: Classification rates achieved on GREC database, with a graph edit distance, with a matricial representation preprocessed or not by a feature selection

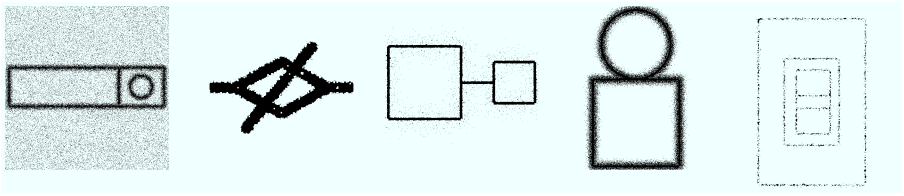

Figure 3: Examples of GREC symbols

Table 2 shows the classification rates claimed with a matricial representation of 295 patterns, with a vector reduced to 42 elements, both compared to the reference. The results highlights several points. The first one is the stability of the representation compared to the graph edit distance. More, the increase of the rate obtained with the vector with selected characteristics shows that some elements of the matrix confuses the classifier because of their fluctuation. These characteristics are omitted with the selection. The second one is the reduce of the computation time. This has not been quantified but is sensitive enough to be noticed. The study of the final vector enhance five of six topological patterns (patterns without attributes), *i.e.* the importance of the topology recognition and the choice of non-local patterns.

3.2 Letter Database

This graph data set involves graphs that represent distorted letter drawings. 15 capital letters of the Roman alphabet are considered (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z). For each class, a prototype line drawing is manually constructed. These prototype drawings are then converted into prototype graphs by representing lines by undirected edges and ending points of lines by nodes. Each node is labeled with a two-dimensional attribute giving its position relative to a reference coordinate system. Edges are unlabeled. The graph database consists of 2250 graphs uniformly distributed over the 15 classes. In order to test classifiers under different conditions, distortions are applied on the prototype graphs with three different levels of strength, low, medium and high. Hence, the total data set comprises 6,750 graphs altogether. In Fig. 4 the prototype graph and a graph instance for each distortion level representing the letter A are illustrated. The authors of [9] achieved classification rates of 99.6% (low), 94.0% (medium), and 90.0% (high).

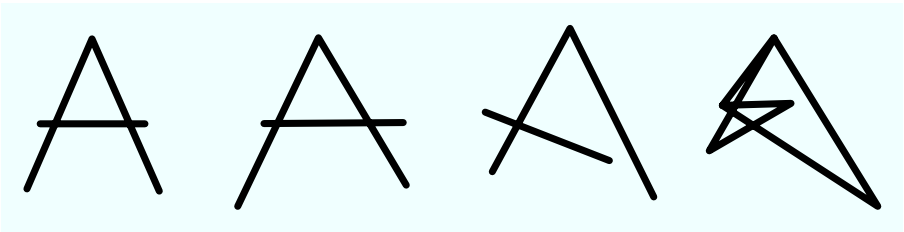

Figure 4: Examples of differents distortion of the letter A

Graph edit distance	60 patterns	13 patterns
99.60 %	91.86 %	92.53 %

Table 3: Classification rates achieved on Letter database, with a graph edit distance, with a matricial representation preprocessed or not by a feature selection

Table 3 presents the classification results obtained with a graph edit distance, the whole matricial representation of 60 patterns and a preprocessed one with 13 patterns. The rates shows a small evolution between the 2 vectors, but the smaller size of the second one improves the time to classify the graphs. Unlike the first

database, the topological patterns are less representative in this set. The highlighted features are the pattern one edge with two vertices attributed with all the combination of attributes. In this set, attributes are more relevant than topology. This can explain that rates are inferior to the graph edit distance because of the discretization of numerical data.

4 Conclusions

An answer to solve the complexity problem due to the computation of the distance between 2 graphs can be the graph embedding. This article presented a method to represent the graph topology associated with its label in a numerical matrix.
This matrix allows the use of known and very used tools. Many distances can be applied to have a classification processed in a linear complexity, for example a euclidean distance in this article. More, we presented, the improvment of the occurence matrix by selecting specific feature to specialize the signature to the data. The results are hopeful and encourage the pursuit of our works.
Our future aims will be the quantification of the gain of time in the computation of distances, the improvment of the discretization of numerical labels and to use the occurence matrix to retrieve the occurences of a subgraph in a graph.

Acknowledgments

The works described in this article were done with the support of the ANR within the project NAVIDOMASS ANR-06-MDCA-12 and with the participation of the regional council “Centre” within the project PIVOAN.

References

[1] Conte, D. and Foggia, P. and Sansone, C. and Vento, M. : Thirty Years Of Graph Matching In Pattern Recognition, International Journal of Pattern Recognition and Artificial Intelligence, 2004

[2] Garey, M.R. and Johnson, D.S. : Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York, 1979.

[3] Ullmann, J.R.: An algorithm for subgraph isomorphism. J. ACM, 1976

[4] Cordella, L., Foggia, P., Sansone, C., Vento, M. : An improved algorithm for matching large graphs, 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, 2001.

[5] Lopresti, D.,Wilfong, G. : A fast technique for comparing graph representations with applications to performance evaluation, Int. J. Doc. Anal. Recognit., vol. 6, nb. 4, 2003

[6] Jaromczyk, J., Toussaint, G. : Relative neighborhood graphs and their relatives Proceedings of the IEEE, 1992

[7] Barbu, E., Heroux, P., Adam, S., Trupin, E. : Clustering document images using a bag of symbols representation, ICDAR, 2005

[8] Sidere, N., Heroux, P., Ramel, J.Y. : A Vectorial Representation for the Indexation of Structural Informations, S+SSPR, 2008

[9] Riesen, K., Bunke H. : IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning, S+SSPR 2008

[10] Dosch, P., Valveny, E.: Report on the second symbol recognition contest. GREC 2005.

[11] Watson, C., Wilson, C.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology 1992

[12] Neuhaus, M., Bunke, H.: A graph matching based approach to fingerprint classification using directional variance. AVBPA 2005

[13] U.M. Fayyad and K.B. Irani : Multi-interval discretization of continuous-valued attributes for classification learning. Morgan Kaufman, 1993.

[14] Hall M.A. : Correlation-based Feature Subset Selection for Machine Learning. Hamilton, New Zealand. 1998

[15] Sidere N., Heroux P., Ramel J.Y : Vector Representation of Graphs : Application to the Classification of Symbols and Letters. ICDAR 2009.